ChinaXiv合作期刊

具有学习机制的正弦余弦算法 *

方旭阳，武相军†，游大涛

(河南大学 软件学院，河南 开封 475004)

摘　要：针对标准正弦余弦算法在求解函数优化问题时易陷入局部最优、收敛精度较差等问题，提出了一种具有学习机制的正弦余弦算法。该算法引入精英反向学习策略构造精英及反向群体，对其混合群体进行择优保留，从而优化了种群中的个体位置、提高了算法的寻优精度；同时，利用个体的反思学习能力来防止个体盲目地向当前最优解学习，使算法停滞在局部最优，从而有效地避免了算法的未成熟收敛。通过 13 个标准测试函数进行仿真实验，实验结果证明，该算法相比于对比算法具有较强的鲁棒性和函数优化能力。

关键词：正弦余弦算法；精英反向学习；群体智能；反思学习

中图分类号：TP301.6　　doi: 10.3969/j.issn.1001-3695.2018.08.0620

Sine cosine algorithm with learning mechanism

Fang Xuyang, Wu Xiangjun+, You Datao

(*College of software*, *Henan University*, *Kaifeng Henan* 475004, *China*)

Abstract: In order to solve function optimization problems that standard sine cosine algorithm is easy to fall into local optimum and poor convergence accuracy, the paper proposed a sine cosine algorithm with learning mechanism. The algorithm introduced the elite opposite learning strategy to construct the elite and the opposite population, and preserve the optimal individual in the mixed population, thus optimize the individual position in the population and improve the optimization accuracy of the algorithm. At the same time, the algorithm used individual reflective learning ability to avoid individuals blindly learning from the current optimal solution, and made the algorithm stagnating in the local optimal, thus effectively avoid the immature convergence of the algorithm. The algorithm performed simulation experiments with 13 standard test functions. The experimental results showed that the algorithm was more robust and better function optimization ability than comparison algorithms.

Key words: sine cosine algorithm; elite opposition-based learning; swarm intelligence; reflective learning

chinaXiv:201901.00056v1

0　引言

在科学和工程领域，优化问题一直备受关注。由于问题的复杂性，使用传统方法解决有时是不可能的。目前，许多研究人员一直致力于开发一种新的优化算法，来解决现实中复杂的优化问题。其中群体智能算法应用较为广泛，如粒子群优化（particle swarm optimization，PSO）[1]、人工蜂群算法（artificial bee colony algorithm，ABC）[2]、引力搜索算法（gravitational search algorithm，GSA）[3]和鲸鱼优化算法（whale optimization algorithm，WOA）[4]等。

正弦余弦算法（sine cosine algorithm，SCA）[5]是 Mirjalili 于 2016 年提出的一种新型的群体智能优化算法，该算法结构简单、参数较少且易于实现，它的搜索过程主要受正弦和余弦函数的影响。然而，像其他群体智能优化算法一样，存在收敛精度低、收敛速度慢、易陷入局部最优等问题。近两年来，许多研究人员从不同的角度提出了不同的改进方法，以提高算法的优化能力。文献[6]提出了一种转换参数非线性递减的正弦余弦算法，分别以抛物线函数和指数函数控制参数 r_1 的变化，实验结果表明指数函数对参数 r_1 的调整，能够较好的平衡算法的全局勘探和局部开发能力。文献[7]提出了量子计算与正弦余弦算法相结合的方法，利用量子比特对位置进行编码、量子旋转门对个体状态进行更新、量子非门实现变异操作等，实验验证了算法的有效性，达到较好的效果。文献[8]引入反向学习的方法，对当前个体产生反向解，扩大了对解空间的探索。文献[9]提出了混合灰狼优化的正弦余弦算法，利用正弦余弦更新公式改进头狼的移动方向和速度，使算法得到较好的提升。文献[10]提出了结合差分进化的正弦余弦算法，使用差分进化算子帮助算法跳出局部最优解区域。

鉴于此，为了提高 SCA 算法的收敛速度和避免早熟现象，本文在基本的 SCA 算法上提出了精英反向学习策略与反思学习策略相结合的改进方法。通过对标准测试函数的仿真实验，并与基本的 SCA、PSO、WOA 及其几种改进的 SCA 算法相比较，实验表明本文所改进的算法显著提高了算法的全局优化能力。

1　正弦余弦算法

SCA 算法是最近提出的一种新颖的全局优化算法，有别于根据不同生物启发机理的群体智能优化算法。在 SCA 算法中，个体的更新状态主要依赖于两个数学函数 Sine 和 Cosine

收稿日期：2018-08-15；**修回日期：**2018-10-10　**基金项目：**国家自然科学基金资助项目（61872125）；河南省科技厅资助项目（182102410051）；赛尔网络下一代互联网技术创新项目（NGII20170902）

作者简介：方旭阳（1994-），男，河南信阳人，硕士研究生，主要研究方向为机器学习、智能优化；武相军（1980-），男（通信作者），河南安阳人，教授，博士，主要研究方向为信息安全、信号处理、金融大数据处理(wuhsiang@yeah.net)；游大涛（1981-），男，河南周口人，副教授，博士，主要研究方向为语音信号处理、机器学习、智能优化.

ChinaXiv合作期刊

的变化来实现优化搜索。假设第 t 代种群中个体 X_i^t 由 D 个分量组成，即 $X_i^t=(x_{i1}^t, x_{i2}^t,\ldots, x_{id}^t)$， $i\in\{1,2,\ldots, N\}$，$d\in\{1,2,\ldots, D\}$，N 为种群规模，D 为个体维度。种群的初始化使用式（1）随机产生个体所在位置。

$$X_i^d = X_{min}^d + rand(0,1)(X_{max}^d - X_{min}^d) \tag{1}$$

其中：X_{max}^d和X_{min}^d 表示个体在第 d 维度的上下界。

在每次迭代中，第 i 个个体的位置，将按以下更新方程移动所在位置，即

$$X_i^{t+1}=\begin{cases} X_i^t + r_1\times sin(r_2)\times|r_3P_{best}-X_i^t| & r_4<0.5 \\ X_i^t + r_1\times cos(r_2)\times|r_3P_{best}-X_i^t| & r_4\geq 0.5 \end{cases} \tag{2}$$

其中：X_i^t 表示在第 t 次迭代中个体 i 的位置；P_{best} 表示种群的当前最优位置；r_2，r_3，r_4 是服从均匀分布的随机数，$r_2\in[0,2\pi]$，$r_3\in[-2,2]$， $r_4\in[0,1]$， r_1 是控制参数。

如式（2）所示，式中有四个主要的调节参数分别是 r_1、r_2、 r_3 和 r_4。参数 r_1 决定下一次迭代时第 i 个个体的位置区域，定义了下次迭代的最大可能移动的区域范围；r_2 定义了当前解应该朝目标解前进还是远离；参数 r_3 是目标解的一个随机权值，当 $|r_3|>1$ 增强目标解对当前解的影响，当 $|r_3|<1$ 目标解对当前解的影响减弱；r_4 是正弦和余弦机制切换的随机概率。 $r_1\sin(r_2)$ 或 $r_1\cos(r_2)$ 共同领导着算法进行局部搜索和全局搜索，当 $r_1\sin(r_2)$ 或 $r_1\cos(r_2)$ 的值大于 1 或小于-1 时，进行全局勘探；在-1 与 1 之间时，进行局部开发。为了使算法的勘探和开发能力得到平衡，寻找到解空间中可能的区域，并最终收敛到全局最优， r_1 通过式（3）进行自适应调整，即

$$r_1 = a - t\frac{a}{T} \tag{3}$$

其中： t 是当前迭代的次数； T 是最大迭代次数； a 是一个常数。在算法中， a 一般设置为 2。

SCA 算法通过数学函数的变化来改变初始状态的位置，种群中的个体更新依靠函数值的增加或者减少来随机地更新每次迭代中每个个体的状态，使种群在前期保持多样性，后期随着 r_1 的减少，个体趋于局部开发，最终收敛于全局最优状态。算法的基本原理如图 1 所示。

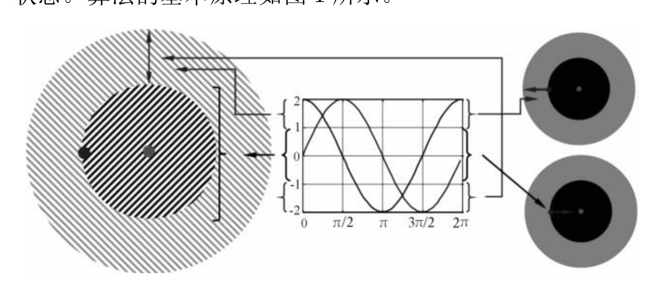


图 1　正弦余弦算法的基本原理

Fig. 1　Basic principle of sine-cosine algorithm

2　改进的正弦余弦算法

2.1　精英反向学习

SCA 算法在迭代后期将对当前全局最优位置附近进行很小的邻域内搜索，不断尝试更新最优解。如果搜索过程远离了理论最优解，算法就很难在短时间内收敛到全局最优。因此本文引入精英反向学习策略，扩大精英种群的搜索范围，从而改善了 SCA 算法收敛速度慢、精度不足等问题。

2.1.1 基本概念

反向学习（opposition-based learning）[11~12] 策略是近年来智能优化邻域出现的一种新技术，目前已在人工蜂群算法（ABC）、差分进化算法（DE）、粒子群算法（PSO）等群体智能优化算法中得到了成功的应用。它的主要思想是：每个个体在搜索空间中存在一个反向个体，反向个体可能更接近于最优位置。这样就能在每次搜索过程中，同时搜索当前个体和当前个体的反向解，对其进行择优保留。保留下来的优秀个体加速了种群中个体向最优位置靠近，提高了算法的求解效率。

定义 1　反向解(opposite solution)[12]。设在区间[lb,ub]上存在一个实数 x，则 x 的反向数定义为 $x'=\mathrm{lb}+\mathrm{ub}-\mathrm{x}$。鉴于此，假设在 R 域上存在一个点 X 在 D 为的空间上，则 $\mathrm{X}=(x_1,x_2,\ldots,x_D)$，并且 $x_i\in[lb_i,ub_i]$，即它的反向解 $\mathrm{X}'=(x_1',x_2',\ldots,x_D')$,由式 $x_i'=k*(lb_i+ub_i)-x_i$生成，k 为[0,1]之间均匀分布随机数，称为一般化系数。

定义 2　基于反向解的优化(opposite-based optimization)[12]。设存在种群 P，其反向种群为 P',基于优者保留的方式从 $\mathrm{P}\cup P'$ 的种群中选择最优秀的 n 个个体体组成新种群，称为基于反向解的优化。

定义 3　动态一般反向学习(dynamic generalized opposition-based learning)[12]。设存在种群 X，数量为 N，维度为 D，则第 t 次迭代过程中，种群的反向解为

$X_{ij}'=k*(lb_j^t+ub_j^t)-X_{ij}^t$，其中 $lb_j^t=\min(X_{ij}^t)$, $ub_j^t=\max(X_{ij}^t)$。 X_{ij}^t 为

种群第 i 个体在第 j 维上的分量。

定义 4　精英反向解(elite opposition solution)[12]。设在 D 维空间上 $\mathrm{X}_{best}'=(x_1',x_2',\ldots,x_D')$ 为当前种群的精英个体 $\mathrm{X}_i^{best}=(x_1,x_2,\ldots,x_D)$ 的反向解，该反向解定义为 $x_i'=k*(lb_i+ub_i)-x_i$, $\mathrm{k}\in[0,1]$ 为服从均匀分布的随机数，利用该系数可以生成精英个体的多个反向解。

通过生成的精英反向解，增加了种群向全局最优收敛的有益信息，可以加强对最优个体周围邻域的探索，提高算法的局部开发能力。

2.1.2 精英反向策略

为了减少算法偏离全局最优位置的可能性，加强对优秀个体周围空间的搜索是非常有必要的，这种改进可以提高算法勘探新解的能力。本文将精英反向学习的策略融入 SCA 算法中，充分利用精英种群的信息，搜索精英个体及其反向解所在的空间。具体操作如下：

种群中的个体由式（2）执行后，对种群进行排序，取其中 10%的优秀个体组成精英种群 P_{best}。

计算个体 $\mathrm{X}_{best}^i\in P_{best}^t$ 的边界 $[lb_j^i,ub_j^i]$,并求解其动态边界 $[\min(lb_j^i),\max(ub_j^i)]$。

根据定义 3、4 生成个体 X_{best}^i 的动态精英反向种群 P_{best}'。

若反向种群 P_{best}' 超出了 X_{best}^i 的动态边界 $[\min(lb_j^i),\max(ub_j^i)]$ 的限制，则由式（1）在边界内随机生成新个体进行替换。

将混合种群 $P_{best}\cup P_{best}'$ 根据适应度大小从高到低排序，选择前 1/2 的个体进入下一代。

循环执行 b ~ e,直到达到停止条件，算法结束。

2.2　反思学习策略

在教与学优化算法[13~15]中，所有的学生通过教师的课堂教学学习科目知识，同时还向身边同学请教来不断提高自己的学习能力。文献[16]提出基于自主学习行为的教与学优化算法，把学习分类为自主学习阶段和反思阶段。由于学生个体无法充分发挥自身的学习能力，最终影响整体的学习效果，所以在求解复杂函数优化问题时，容易过早陷入局部最优区域。受此启发，本文对 SCA 算法进行以下改进，提升个体的学习能力。

在 SCA 算法中，种群的个体只依赖于当前最优解来进行自身状态的更新。因此，算法有较大的可能陷入局部最优状

chinaXiv:201901.00056v1

ChinaXiv合作期刊

态，导致了算法无法寻找到满意的解。此时，需要对个体进行局部变异操作，个体以当前的学习结果向前一次的学习情况进行反思学习，增加逃逸局部区域的概率。反思学习的公式如（4）所示，即

$$X_i^* = X_i^s + \omega \otimes (X_i^s - X_i^t) \tag{4}$$

其中：X_i^t 表示在第 t 次迭代中个体 i 的位置；X_i^s 表示执行式（2）后的位置；X_i^* 表示经过反思过程产生的新位置；ω 表示学习因子，$\omega \in [-1,1]$；$\otimes$ 表示点乘。

每一代个体在执行正弦或余弦操作后，开始进入自我反思阶段。根据自身前一次的学习状态反思当前的学习状态，对当前学习状态进行自我修正。通过自身学习的方式能够更好的防止个体易受当前全局最优解影响，而陷入局部最优。这里为了防止反思过程中的随机性太大，将用式（5）对学习因子 ω 进行约束，同时为了避免学习能力退化、增强算法的收敛性，使用贪婪学习的方式，个体在反思前后的学习状态根据学习成绩（适应度）进行择优选择。

$$\omega = C^{(-t/T)} \times \cos(r_5) \tag{5}$$

其中：r_5 是 $[0,\pi]$ 上的随机数; C 是常数，经实验 C=100 时效果最好。

学习因子随迭代次数变化情况如图 2 所示。

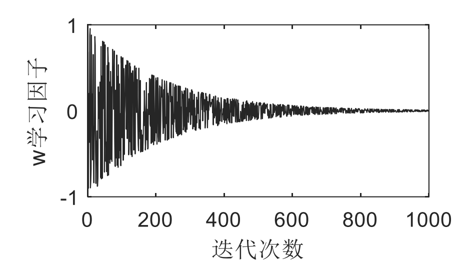


图 2　学习因子随迭代次数变化情况

Fig. 2　Learning factor changes with number of iterations

由图 2 可以看出，$C^{(-t/T)}$ 在迭代过程中非线性递减，即个体在前期具有较强的反思活动，实现自我的超越，提高勘探能力；在后期随着自身状态的不断提高，趋于全局最优状态，反思活动不断减少，学习因子 ω 的强度逐渐变弱。

在式（6）中，个体在通过反思后，若优于反思前的个体将对该个体进行替换。使用贪婪选择的方式将最优个体送入下一代，确保了每一代个体都趋向于全局最优解方向移动。

$$X_i^{t+1} = \begin{cases} X_i^s & f(X_i^s) \le f(X_i^*) \\ X_i^* & Otherwise \end{cases} \tag{6}$$

其中：$f(\cdot)$ 表示个体的适应度。

反思学习策略能够较大地提高算法的全局优化能力，个体对自己当前的学习状态进行深入思考，并根据上次学习状态进行变异性调整，有效地防止了自己停滞在局部状态，提高了规避局部最优的能力。

2.3　改进的算法步骤

为了便于后续的实验对比分析改进后的算法记为 MSCA(modified sine cosine algorithm)。改进后的算法以伪代码来表示，算法的具体流程如表 1 所示。

MSCA 算法：

初始化参数，利用式（1）随机产生种群数量为 N 的个体

$X_i = (X_{i1}, X_{i2}, \dots, X_{iD})$；

计算每个个体的适应度；

设置最优解的位置及适应度；

do

　for i=1 to N

　　更新 r_1, r_2, r_3, r_4

　　if r_4 <0.5

　　　$X_i^s = X_i^t + r_1 \times \sin(r_2) \times |r_3 P_{best} - X_i^t|$

　　else

　　　$X_i^s = X_i^t + r_1 \times \cos(r_2) \times |r_3 P_{best} - X_i^t|$

　　正弦余弦操作后产生的个体的新位置进行反思学习：

　$X_i^* = X_i^s + \omega \otimes (X_i^s - X_i^t)$

　　选择反思前后适应度最好的个体作为下一代：

$$X_i^{t+1} = \begin{cases} X_i^s & f(X_i^s) \le f(X_i^*) \\ X_i^* & \text{Otherwise} \end{cases}$$

　end for

　对当前种群的前 10%的最优个体计算反向位置：

$$X_{ij}^{'} = rand * (lb_j^t + ub_j^t) - X_{ij}^t$$

　选择 1/2 的 $P_{best} \cup P_{best}^{'}$ 混合群体中的最优个体；

　计算种群适应度,更新最优解位置及适应度；

while（current iteration < maximum iterations）

return best solution obtained as the global optimum

MSCA 算法主要由初始化、正弦余弦位置更新、反思学习、精英反向学习四部分组成。由上述算法的流程可得，假设算法的迭代次数为 m，种群数量为 n,维度为 d，其中种群初始化和个体的位置更新与基本的 SCA 算法复杂度相同，为 $O(mnd)$ 。在反思学习过程中，对每个个体进行位置扰动，其时间复杂度为 $O(mnd)$；在精英反向学习的过程中，需要对 10%的个体进行反向计算且构造混合群体进行比较，其时间复杂度为 $O(mnd)$ 。因此，整个算法的时间复杂度为 $O(mnd)$ 。

3　实验仿真

3.1　实验环境及测试函数

为了验证 MSCA 算法对函数的优化能力，本文采用文献[9]中的 13 个标准测试函数进行实验对比分析。仿真实验运行环境为 Intel Core i7 CPU，主频 3.60 GHz，内存 8 GB，Windows7 64 位操作系统，实验仿真软件采用 MATLAB R2017b。实验中采用的算法统一设置为种群规模 N=30，最大迭代次数 T=1000，算法在每个标准测试函数上独立运行 30 次，统计其平均值和标准差。对比的算法和描述如表 1 所示。测试函数如表 2、3 所示。

表 1　对比算法及描述

Table 1　Contrast algorithms and descriptions

算法	描述
SCA	标准正弦余弦算法
OBSCA[8]	基于反向学习的正弦余弦算法
HGWOSCA[9]	混合灰狼优化的正弦余弦算法
WOA	标准鲸鱼优化算法
PSO	标准粒子群算法

其中：PSO 的参数设置为 $C_1=2$，$C_2=2$，$W_{max}=0.9$，$W_{min}=0.2$ 。

表 2　单峰基准测试函数

Table 2　Unimodal benchmark functions

Fun	函数名	搜索区间	理论最优值
$F1$	Sphere	[-100,100]	0
$F2$	Schwefel 2.22	[-10,10]	0
$F3$	Schwefel 1.2	[-100,100]	0
$F4$	Schwefel 2.21	[-100,100]	0
$F5$	Rosenbrock	[-30,30]	0
$F6$	Step	[-100,100]	0
$F7$	Quartic Noise	[-1.28,1.28]	0

chinaXiv:201901.00056v1

chinaXiv:201901.00056v1

表 3　多峰基准测试函数

Table 3　Multimodal benchmark functions

Fun	函数名	搜索区间	理论最优值
*F*8	Schwefel 2.26	[-500,500]	-418.9829*n
*F*9	Rastrigin	[-5.12,5.12]	0
*F*10	Ackley	[-32,32]	0
*F*11	Griewank	[-600,600]	0
*F*12	Penalized1	[-50,50]	0
*F*13	Penalized2	[-50,50]	0

3.2　实验结果分析

为了体现本文算法在问题维数和复杂度上的优越性，实验将分别对维数为 30、100 两种不同的函数维度进行测试。其中 F1～F7 为连续性单峰值函数，F8～F13 为连续性多峰函数，均是具有普遍性、代表性的常用于检测群体智能算法全局优化能力的函数。函数的表达式可以查阅文献[9]。

1）寻优精度分析

表 4 列出了六种算法在单峰值函数求解时 30 次运行结果的平均值和标准差的对比。由表 4 可见，MSCA 算法在 F1 函数上的求解精度高出 WOA 算法及其他改进的正弦余弦算法 80～200 个数量级；对 F3、F4 函数的求解精度较对比算法效果十分显著；在 F6 函数的求解上，MSCA 算法在高维的求解能力优于较好的 PSO 算法，在低维的求解上较 PSO 算法结果差；在 F5、F7 函数上，算法对 30、100 维的求解精度较低，但 MSCA 算法较其他五种算法仍具有优势；故总体考虑，MSCA 在单峰函数问题的求解精度、算法稳定性（标准差）等方面比对比算法具有更好的效果。

表 5 列出了六种算法在多峰值函数求解时 30 次运行结果的平均值和标准差的对比。由表 5 可以看出，对于 F8～F13 多峰值函数的寻优结果，在问题维度为 30、100 维时 MSCA 均优于对比算法；在 F8 函数的求解上，相比于对比算法，MSCA 能够找到接近理论最优解的结果，SCA 在该函数上的处理能力最差；在 F9、F11 函数的求解上，WOA、MSCA 算法能达到理论最优值且 MSCA 在 30、100 维求解性能较为稳定；在 F10、F12、F13 函数的求解上，MSCA 的寻优结果与理论最优解相比还存在精度不足，但优于对比算法。可见 MSCA 在多峰值函数上的优化能力比较强，逃逸局部最小值的能力相比而言要好于对比算法。故总体考虑，MSCA 对多峰函数问题的求解表现具有很大的优越性。

2）收敛曲线分析

为了进一步考查算法的平均进化趋势，实验中列出了 30 维时算法的收敛曲线。限于篇幅限制，在文中仅列出了 F1、F5、F6、F7 单峰测试函数，F9、F10、F12、F13 多峰测试函数的平均函数值的迭代收敛曲线。部分单峰值函数的收敛曲线如图 3 所示。由图 3（a）～（d）可以看出，MSCA 算法的收敛曲线下降最快，并能够随着迭代次数的增加持续收敛，未出现停滞现象。其中 WOA 算法是较新的优化算法，它在单峰值函数的求解精度相比其他群体智能算法要好。通过图中显示，MSCA 算法在 F1、F5、F7 函数上收敛趋势优于 WOA 算法，也优于其他改进的 SCA 算法。从图 3（b）可知，对比算法在后期收敛出现停滞，收敛精度较低，而 MSCA 算法能规避停滞发生。由图 3（c）可看出，WSCA 与 PSO 相比在后期迭代效果不佳，前期收敛下降较快，但优于图中的对比算法。

图 4（a）～（d）是部分多峰值函数的收敛曲线。从 MSCA 算法的收敛性上可以看出，在 F9 函数上收敛速度明显，在 F10、F12 和 F13 函数上，也比对比算法收敛程度好。由图 4（b）～（d）F10、F12 和 F13 函数的收敛结果可以看出，该函数易陷入局部最优，但 MSCA 最终的寻优精度较其他算法仍然表现较好。

表 4　函数 F1～F7 的寻优精度对比

Table 4　Variance and mean information of six algorithms(F1～F7)

Fun	算法	30 维			100 维		
		平均值	标准差	排序	平均值	标准差	排序
F1	WOA	1.3535E-150	4.7989E-150	2	7.9419E-147	4.3008E-146	2
	OBSCA	7.0799E-28	2.6100E-27	4	8.0438E-09	2.6023E-08	4
	HGWOSCA	1.3165E-50	1.8525E-50	3	1.6487E-25	1.4927E-25	3
	SCA	6.2969E-03	1.0909E-02	6	4.7299E+03	3.5584E+03	6
	PSO	8.4698E-09	2.0700E-08	5	2.5551E+00	1.4886E+00	5
	MSCA	**1.2395E-233**	**0.0000E+00**	**1**	**2.5194E-220**	**0.0000E+00**	**1**
F2	WOA	2.7529E-103	1.0638E-102	2	7.0937E-103	3.1981E-102	2
	OBSCA	5.1910E-26	1.6873E-25	4	2.4920E-13	6.7044E-13	4
	HGWOSCA	6.9748E-30	1.0040E-29	3	1.2850E-15	5.0447E-16	3
	SCA	1.6508E-05	2.8543E-05	5	2.7108E+00	4.2881E+00	5
	PSO	3.5512E-04	8.2176E-04	6	1.1212E+01	4.1496E+00	6
	MSCA	**9.7071E-108**	**5.3011E-107**	**1**	**2.4938E-106**	**1.3045E-105**	**1**
F3	WOA	2.3101E+04	1.0649E+04	6	9.0586E+05	1.7474E+05	6
	OBSCA	2.5799E-03	1.2901E-02	3	7.3762E+03	6.8831E+03	3
	HGWOSCA	1.1660E-12	3.7569E-12	2	1.9520E+01	4.2611E+01	2
	SCA	4.6954E+03	3.3049E+03	5	1.9059E+05	5.3055E+04	5
	PSO	1.6050E+01	8.5760E+00	4	1.0830E+04	2.4641E+03	4
	MSCA	**9.0933E-151**	**4.8622E-150**	**1**	**4.3847E-114**	**2.2422E-113**	**1**
F4	WOA	3.1947E+01	2.8662E+01	6	7.0368E+01	2.7823E+01	5
	OBSCA	7.4303E-05	1.5470E-04	3	3.8343E+01	1.1390E+01	4
	HGWOSCA	1.0879E-11	2.3705E-11	2	1.7681E-03	2.2282E-03	2
	SCA	2.0936E+01	1.1576E+01	5	8.5688E+01	3.3002E+00	6
	PSO	6.2625E-01	1.6545E-01	4	8.3142E+00	1.1723E+00	3
	MSCA	**1.7361E-113**	**9.0691E-113**	**1**	**9.8756E-112**	**5.3552E-111**	**1**
F5	WOA	2.7015E+01	4.0893E-01	2	9.7582E+01	4.3771E-01	2
	OBSCA	2.8197E+01	2.5030E-01	4	9.9191E+01	7.8873E-01	4
	HGWOSCA	2.7151E+01	9.1918E-01	3	9.7905E+01	6.2451E-01	3
	SCA	7.3237E+02	2.9100E+03	6	5.3408E+07	2.3688E+07	6
	PSO	4.9549E+01	3.1024E+01	5	2.4044E+03	1.3275E+03	5
	MSCA	**1.9731E-05**	**2.9621E-05**	**1**	**5.5978E-05**	**9.1330E-05**	**1**
F6	WOA	5.7319E-02	6.8223E-02	3	2.0318E+00	7.9950E-01	2
	OBSCA	4.3876E+00	1.8999E-01	5	2.2373E+01	4.6430E-01	5
	HGWOSCA	1.1532E+00	4.4087E-01	4	1.0407E+01	1.1248E+00	4
	SCA	4.7120E+00	4.4180E-01	6	6.9219E+03	3.9950E+03	6
	PSO	5.2557E-09	8.2397E-09	1	2.7054E+00	1.2978E+00	3
	MSCA	**5.0650E-08**	**9.6047E-08**	**2**	**2.1809E-07**	**3.5146E-07**	**1**
F7	WOA	2.1976E-03	2.5725E-03	3	1.5681E-03	1.5729E-03	2
	OBSCA	2.4692E-03	1.5650E-03	4	1.0853E-02	6.2737E-03	4
	HGWOSCA	1.0902E-03	5.9821E-04	2	3.6505E-03	1.3202E-03	3
	SCA	2.8870E-02	2.3546E-02	5	6.6319E+01	4.0981E+01	5
	PSO	6.7384E-02	2.7413E-02	6	1.4099E+03	3.6454E+02	6
	MSCA	**3.5682E-05**	**3.2475E-05**	**1**	**4.3021E-05**	**3.8397E-05**	**1**

以上收敛图表明了 MSCA 算法可以以更快的收敛速度得到精度更高的优化结果，在六种算法中全局搜索能力最佳。

3）综合实验分析

仿真实验的算法排序结果如表 6 所示。表中列出了六种算法在 13 个标准测试函数的 30、100 维度的平均排序结果及总排序。在同等的实验前提下，MSCA 算法相比于对比算法

ChinaXiv合作期刊

chinaXiv:201901.00056v1

具有显著优势。

表 5　函数 F8～F13 的寻优精度对比

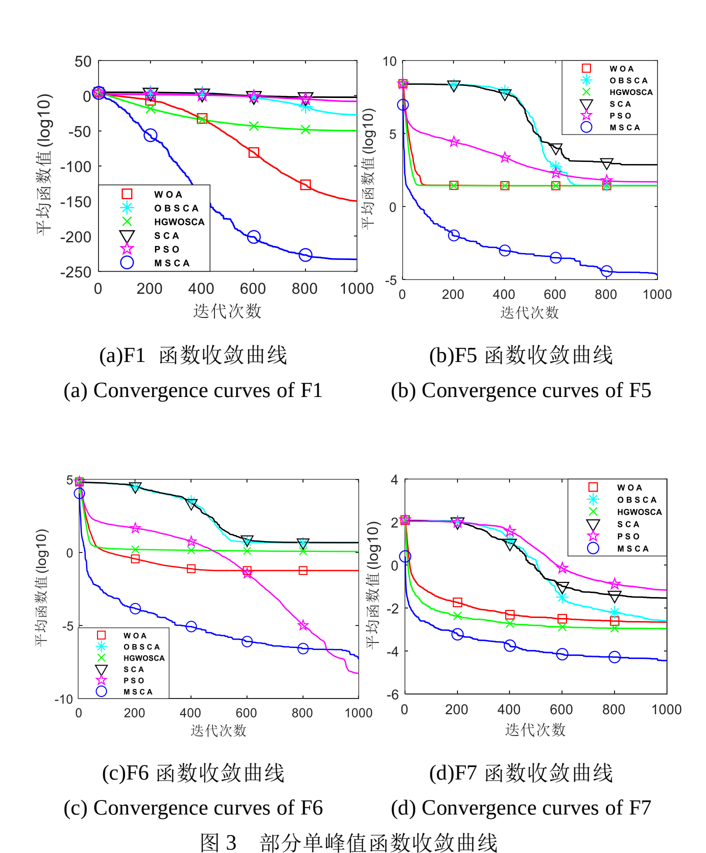
Table 5　Variance and mean information of six algorithms(F8～F13)

Fun	算法	30 维 平均值	30 维 标准差	30 维 排序	100 维 平均值	100 维 标准差	100 维 排序
F8	WOA	-11549.179	1.2447E+03	2	-3.7602E+04	4.5384E+03	2
	OBSCA	-4168.9017	3.1396E+02	5	-7.4608E+03	4.9664E+02	5
	HGWOSCA	-5827.5	7.4939E+02	4	-1.6342E+04	1.7700E+03	4
	SCA	-3998.299	3.3629E+02	6	-7.2570E+03	4.6787E+02	6
	PSO	-6451.3732	1.0114E+03	3	-1.8731E+04	6.1457E+03	3
	MSCA	**-12569.483**	**5.7187E-03**	**1**	**-4.1898E+04**	**1.1390E-02**	**1**
F9	WOA	0.0000E+00	0.0000E+00	1	0.0000E+00	0.0000E+00	1
	OBSCA	0.0000E+00	0.0000E+00	1	2.7440E-06	1.4944E-05	2
	HGWOSCA	5.3355E+00	7.8261E+00	2	1.1795E+01	8.2861E+00	3
	SCA	1.0882E+01	1.5187E+01	3	2.3894E+02	1.0306E+02	4
	PSO	4.9957E+01	1.0100E+01	4	4.5198E+02	5.4017E+01	5
	MSCA	**0.0000E+00**	**0.0000E+00**	**1**	**0.0000E+00**	**0.0000E+00**	**1**
F10	WOA	3.9672E-15	2.7572E-15	2	3.3751E-15	2.3137E-15	2
	OBSCA	2.1618E-02	9.0653E-02	5	4.7178E+00	3.7659E+00	5
	HGWOSCA	2.7060E-14	3.6712E-15	3	2.7930E-13	3.3242E-14	3
	SCA	1.5319E+01	8.0031E+00	6	1.9520E+01	3.3554E+00	6
	PSO	6.1330E-05	7.4154E-05	4	2.6409E+00	3.2792E-01	4
	MSCA	**8.8818E-16**	**0.0000E+00**	**1**	**8.8818E-16**	**0.0000E+00**	**1**
F11	WOA	2.3940E-03	1.3113E-02	3	0.0000E+00	0.0000E+00	1
	OBSCA	3.2839E-13	1.7985E-12	2	4.0223E-08	1.4392E-07	2
	HGWOSCA	2.7258E-03	5.3645E-03	4	5.4231E-03	1.1424E-02	3
	SCA	2.2328E-01	2.8470E-01	6	4.0373E+01	2.8646E+01	5
	PSO	8.6210E-03	1.0422E-02	5	4.8571E-02	3.0360E-02	4
	MSCA	**0.0000E+00**	**0.0000E+00**	**1**	**0.0000E+00**	**0.0000E+00**	**1**
F12	WOA	7.5227E-03	5.9677E-03	3	1.7488E-02	6.5294E-03	2
	OBSCA	4.9258E-01	6.6286E-02	5	1.1934E+00	1.8625E-01	4
	HGWOSCA	7.6204E-02	4.2358E-02	4	3.2058E-01	7.7470E-02	3
	SCA	2.5894E+02	1.0303E+03	6	1.6412E+08	1.0791E+08	6
	PSO	6.9113E-03	2.6302E-02	2	2.2695E+00	1.1890E+00	5
	MSCA	**4.7934E-09**	**7.7293E-09**	**1**	**1.9856E-09**	**3.8542E-09**	**1**
F13	WOA	2.5202E-01	2.0052E-01	3	1.5783E+00	6.7054E-01	2
	OBSCA	2.3644E+00	1.6501E-01	5	1.1341E+01	8.3623E-01	4
	HGWOSCA	9.2799E-01	3.1218E-01	4	6.8008E+00	4.4165E-01	3
	SCA	1.1109E+01	2.4752E+01	6	3.3122E+08	1.8469E+08	6
	PSO	2.5637E-03	4.7266E-03	2	1.3012E+01	7.5262E+00	5
	MSCA	**9.4655E-08**	**1.8124E-07**	**1**	**1.1467E-07**	**1.4137E-07**	**1**

根据实验考查了 MSCA 算法在 30、100 维度下的收敛情况，改善了基本 SCA 算法的性能。基于精英反向学习的方法通过种群间的优胜劣汰保留了更多有利信息，加强了个体对优秀个体的搜索，对提高算法收敛速度和精度具有一定的影响；基于反思学习的方法通过反思变异增加了种群的多样性，防止了个体盲目靠近局部最优；在问题维度上，MSCA 算法不会随着维度的增加而精度下降，表现了良好的稳定性。

4　结束语

本文针对 SCA 算法在搜索策略上存在的不足，提出了一种具有学习机制的正弦余弦算法，能够有效地改善算法求解精度低、未成熟收敛等问题。MSCA 算法利用两种学习机制，弥补了标准算法在全局勘探和局部开发能力上的不足，提高了算法寻找最优解的机会。仿真实验结果表明，MSCA 算法的改进策略是有效的且具有较强的鲁棒性和全局优化能力，有望应用于复杂问题的求解。同时，SCA 作为一种较新的群体智能优化算法具有很大的工程应用空间，有待今后进一步的研究。



(a)F1　函数收敛曲线　　(b)F5 函数收敛曲线

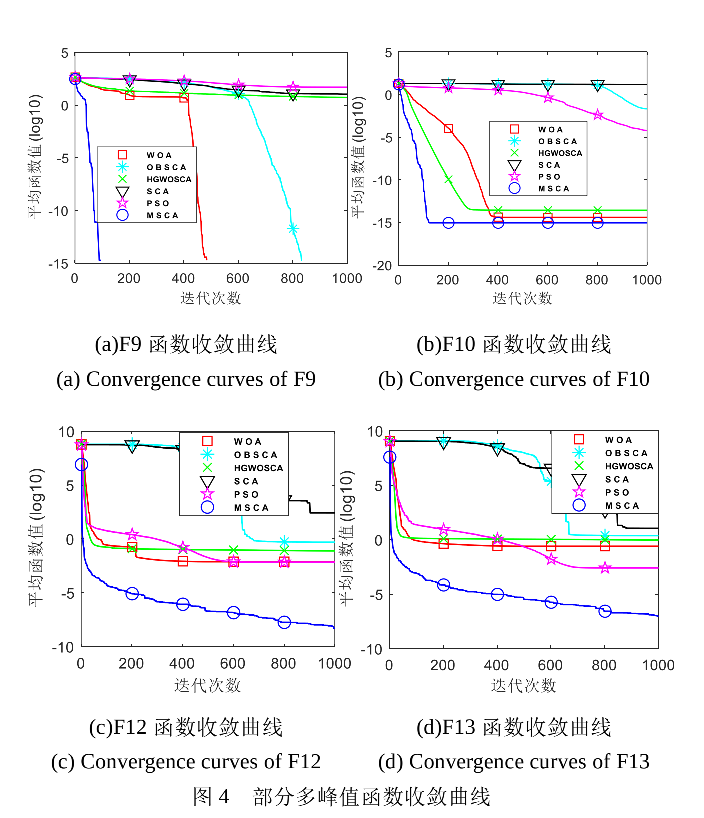
(a) Convergence curves of F1　　(b) Convergence curves of F5

(c)F6 函数收敛曲线　　(d)F7 函数收敛曲线

(c) Convergence curves of F6　　(d) Convergence curves of F7

图 3　部分单峰值函数收敛曲线

Fig. 3　Partial unimodal benchmark functions convergence curve



(a)F9 函数收敛曲线　　(b)F10 函数收敛曲线

(a) Convergence curves of F9　　(b) Convergence curves of F10

(c)F12 函数收敛曲线　　(d)F13 函数收敛曲线

(c) Convergence curves of F12　　(d) Convergence curves of F13

图 4　部分多峰值函数收敛曲线

Fig. 4　Partial multimodal benchmark functions convergence curve

ChinaXiv合作期刊

表 6 算法排序结果

Table 6 Algorithm sorting result

算法	SCA	OBSCA	HGWOSCA	WOA	PSO	MSCA
平均排序	5.50	3.85	3.04	2.65	4.19	1.04
总排序	6	4	3	2	5	1

参考文献：

[1] Verma O P, Gupta S, Goswami S, *et al.* Opposition based modified particle swarm optimization algorithm [C]// Proc of the 8th International Conference on Computing, Communication and Networking Technologies. Delhi: IEEE Press, 2017: 1-6.

[2] 刘三阳，张平，朱明敏. 基于局部搜索的人工蜂群算法 [J]. 控制与决策, 2014 (1): 123-128. (Liu Sanyang, Zhang Ping, Zhu Mingmin. Artificial bee colony algorithm based on local search [J]. Control and Decision, 2014, 29 (1): 123-128.)

[3] 井福荣，郭肇禄，罗会兰，等. 应用精英反向学习的引力搜索算法 [J]. 计算机应用研究, 2015, 32 (12): 3638-3641. (Jing Furong, Guo Zhaolu, Luo Huilan, *et al.* Improved gravitational search algorithm with elite opposition-based learning [J]. Application Research of Computers, 2015, 32 (12): 3638-3641.)

[4] Mirjalili S, Lewis A. The whale optimization algorithm [J]. Advances in Engineering Software, 2016, 95: 51-67.

[5] Mirjalili S. SCA: A sine cosine algorithm for solving optimization problems [J]. Knowledge-Based Systems, 2016, 96: 120-133.

[6] 刘勇，马良. 转换参数非线性递减的正弦余弦算法 [J]. 计算机工程与应用, 2017, 53 (2): 1-5. (Liu Yong, Ma Liang. Sine cosine algorithm with nonlinear decreasing conversion parameter [J]. Computer Engineering and Applications, 2017, 53 (2): 1-5.)

[7] 陈聪，马良，刘勇. 函数优化的量子正弦余弦算法 [J]. 计算机应用研究, 2017, 34 (11): 3214-3218. (Chen Cong, Ma Liang, Liu Yong. Quantum sine cosine algorithm for function optimization [J]. Application Research of Computers, 2017, 34 (11): 3214-3218.)

[8] Elaziz M A, Oliva D, Xiong Shengwu. An improved opposition-based sine cosine algorithm for global optimization [J]. Expert Systems with Applications, 2017, 90: 484-500.

[9] Singh N, Singh S B. A novel hybrid GWO-SCA approach for optimization problems [J]. Engineering Science and Technology, an International Journal, 2017, 20 (6): 1586-1601.

[10] Elaziz M E A, Ewees A A, Oliva D, *et al.* A hybrid method of sine cosine algorithm and differential evolution for feature selection [C]// Proc of the 24th International Conference on Neural Information Processing. Cham Switzerland: Springer Press, 2017: 145-155.

[11] Mahdavi S, Rahnamayan S, Deb K. Opposition based learning: a literature review [J]. Swarm and Evolutionary Computation, 2018, 39: 1-23.

[12] 李俊，汪冲，李波，等. 基于扰动的精英反向学习粒子群优化算法 [J]. 计算机应用研究, 2016, 33 (9): 2584-2587. (Li Jun, Wang Chong, Li Bo, *et al.* Elite opposition-based particle swarm optimization based on disturbances [J]. Application Research of Computers, 2016, 33 (9): 2584-2587.)

[13] Yu Kunjie, Wang Xin, Wang Zhenlei. An improved teaching-learning-based optimization algorithm for numerical and engineering optimization problems [J]. Journal of Intelligent Manufacturing, 2016, 27 (4): 831-843.

[14] Chen Debao, Zou Feng, Li Zheng, *et al.* An improved teaching–learning-based optimization algorithm for solving global optimization problem [J]. Information Sciences, 2015, 297: 171-190.

[15] Rao R V. Application of TLBO and ETLBO algorithms on complex composite test functions [M]// Chapter 3 of the Teaching Learning Based Optimization Algorithm. Cham Switzerland: Springer Press, 2016: 41-51.

[16] 童楠，符强，钟才明. 基于自主学习行为的教与学优化算法 [J]. 计算机应用, 2018, 38 (2): 443-447. (Tong Nan, Fu Qiang, Zhong Caiming. Improved TLBO algorithm based on self-learning mechanism [J] , Journal of Computer Applications, 2018, 38 (2): 443-447.)

chinaXiv:201901.00056v1